Implementation of Stack using Array

THEORY:

Stack is a linear data structure. It follows a particular order in which the operations are performed. The order may be LIFO(Last In First Out) or FILO(First In Last Out).

There are many real-life examples of a stack.

Considering an example where, plates stacked over one another . The plate which is at the top is the first one to be removed and the plate which is at the bottommost position remains in the stack for the longest period of time.

Basic Operations

Stack operations may involve initializing the stack, using it and then de-initializing it.

A stack is used for two primary operations:

1. push() – Pushing/storing an element into the stack.
2. pop() – Removing/accessing an element from the stack.

To use a stack efficiently, we need to check the status of stack9whether empty or full)

Using the following operations:

1. peek() − get the top data element of the stack, without removing it.
2. isFull() − check if stack is full.
3. isEmpty() − check if stack is empty.

At all times, a pointer is maintained to the recently pushed data on the stack. This pointer always represents the top of the stack, hence it is named top. The top pointer provides top(most recently inserted element’s) value of the stack without actually removing it.

ALGORITHM:

1. Declare a 1D array (int stack[size]) with a fixed size.
2. Declare top and initialize to -1.
3. In main function, display a menu with a list of stack operations, making suitable function calls for each operation.

Push

1. Check whether the stack is full where top!=size-1.
2. If true then print “stack overflow”, and terminate the function.
3. If false increment the top to top++ and set the value of stack to a stack[top] = value.

Pop

1. Check whether the stack is empty where top!==-1.
2. If true then print “stack underflow”, and terminate the function.
3. If false set the value of stack to stack[top], and decrement top to top--.

Display

1. Check whether the stack is empty where top!==-1.
2. If true then print “stack underflow”, and terminate the function.
3. If false then declare a value “i” and initialize with the top. Display stack[i] value and decrement i--.

PROGRAM:



